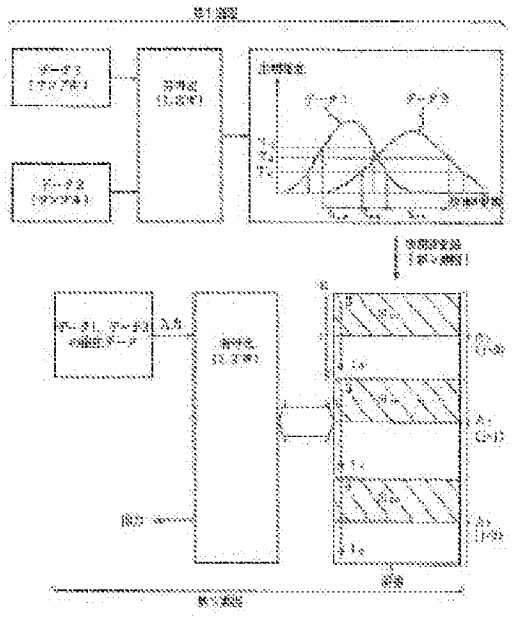
METHOD FOR COMPRESSING PLURAL KINDS OF DATA

Patent number: JP6028149 (A)
Publication date: 1994-02-04
Inventor(s): YOSHIDA SHIGERU; OKADA YOSHIYUKI; NAKANO YASUHIKO; CHIBA HIROTAKA +
Applicant(s): FUJITSU LTD +
Classification:
- international: ***G06F5/00; G06T9/00; H03M7/30; H03M7/46;*** **G06F5/00; G06T9/00; H03M7/30; H03M7/46;** (IPC1-7): G06F15/66; G06F5/00; H03M7/30
- european:
Application number: JP19920183288 19920710
Priority number(s): JP19920183288 19920710

Abstract of **JP 6028149 (A)**

PURPOSE:To provide a method to obtain high compressibility without increasing the processing time of the data compressing for plural kinds of data which compresses and encodes input data by using dynamic dictionary type algorithm of LZW codes by uniting the dictionary retrieval of encoding wherein character string which are investigated as to plural kinds of data and frequently appear are initially registered. CONSTITUTION:Partial strings whose appearance frequency detected by the LZW encoding of sample data 1 and 2 become high in common to plural kinds of data are extracted as a common partial string group S00 and initially registered in a dictionary area A0. Further, partial strings which have high frequencies by the data are extracted as characteristic partial string groups S10 and S20 and initially registered in dictionary areas A1 and A2. When an input character string wherein the data 1 and 2 are mixed is encoded, a registered partial string which matches the input character string to the longest length is retrieved in a dictionary and the input character string is encoded by using the group number (j) of a partial string group belonging to the retrieved partial string and the registration number (i) of the retrieved character string in the partial string group.



(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（A）**　(11)特許出願公開番号

特開平6－28149

(43)公開日　平成６年(1994)２月４日

(51)Int.Cl.[5]	識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０６Ｆ　5/00		Ｈ　9189－5B		
15/66	３３０	Ｈ　8420－5L		
Ｈ０３Ｍ　7/30		8522－5Ｊ		

審査請求　未請求　請求項の数６(全 23 頁)

(21)出願番号　特願平4－183288

(22)出願日　平成４年(1992)７月10日

(71)出願人　000005223
富士通株式会社
神奈川県川崎市中原区上小田中1015番地
(72)発明者　吉田　茂
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内
(72)発明者　岡田　佳之
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内
(72)発明者　中野　泰彦
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内
(74)代理人　弁理士　竹内　進　(外１名)

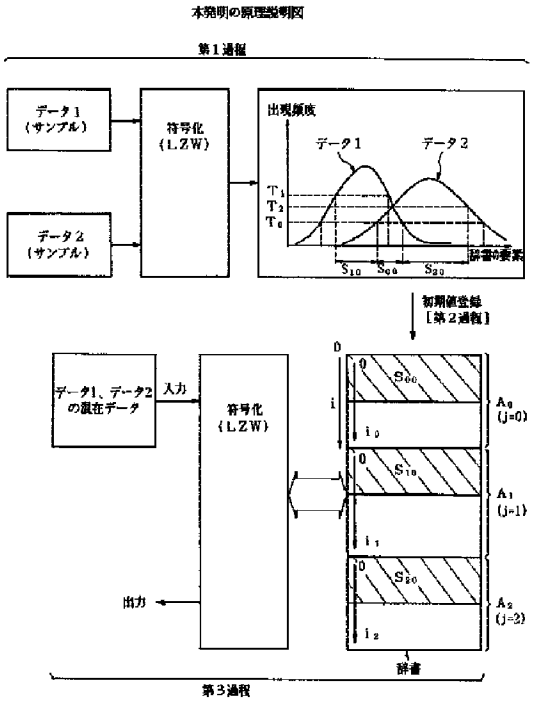
最終頁に続く

(54)【発明の名称】　複数種類データのデータ圧縮方法

(57)【要約】

【目的】ＬＺＷ符号等の動的辞書型アルゴリズムを用いて入力データを圧縮符号化する複数種類データのデータ圧縮方法に関し、複数種類のデータを対象に調べた高頻度に出現する文字列を初期登録した符号化における辞書探索を一本化し、処理時間を増加させることなく高い圧縮率が得られるようにする。

【構成】サンプルデータ１，２のＬＺＷ符号化で検出した出現頻度が、複数種類のデータで共通に高頻度となる部分列を共通部分列群S_{00}として抽出して辞書領域A_0に初期登録する。また複数種類のデータ毎に高頻度となる部分列を固有部分列群S_{10}，S_{20}として抽出して辞書領域A_1，A_2に初期登録する。データ１，２が混在する入力文字列を符号化する際には、入力文字列に最長一致する登録済み部分列を辞書から検索し、検索した部分列の属する部分列群の群番号ｊと部分列群内での検索文字列の登録番号ｉとを用いて入力文字列を符号化する。



1

【特許請求の範囲】

【請求項1】複数種類のデータが混在する入力データを符号化して圧縮する複数種類データのデータ圧縮方法に於いて、

複数種類の各データを相異なる部分列に分けて辞書に登録し、各データ毎に入力文字列に最長一致する辞書に登録済みの部分列を検索し、検索した部分列の登録番号で表わして入力文字列を圧縮符号化し、該符号化における部分列の出現頻度を検出する第1過程と、

前記第1過程で検出した出現頻度が、複数種類のデータで共通に高頻度となる部分列を共通部分列群（S_{00}）として抽出し、該共通部分列群（S_{00}）に特定の群番号（j＝0）を付けて辞書領域（A_0）を確保し、該辞書領域（A_0）に該共通部分列群（S_{00}）に属する各部分列を初期登録し、また前記第1過程で検出した出現頻度が、複数種類のデータ毎に高頻度となる部分列を固有部分列群（S_{10}，S_{20}）として抽出し、各固有部分列群（S_{10}，S_{20}）毎に特定の群番号（j＝1，2）を付けて辞書領域（A_1，A_2）を確保し、該辞書領域（A_1，A_2）に該当する固有部分列群（S_{10}，S_{20}）に属する各部分列を初期登録する第2過程と、

複数種類のデータが混在する入力文字列を符号化する際に、入力文字列に最長一致する登録済み部分列を前記辞書から検索し、検索した部分列の属する部分列群（S_{00}，S_{10}，S_{20}）の群番号（j＝0，1，2）と該部分列群内での検索文字列の登録番号（i_j）とを用いて入力文字列を符号化する第3過程と、を備えたことを特徴とする複数種類データのデータ圧縮方法。

【請求項2】複数種類のデータが混在する入力データを符号化して圧縮する複数種類データのデータ圧縮方法に於いて、

複数種類の各データを相異なる部分列に分けて辞書に登録し、各データ毎に入力文字列に最長一致する辞書に登録済みの部分列を検索し、検索した部分列の登録番号で表わして入力文字列を圧縮符号化し、該符号化における部分列の出現頻度を検出する第1過程と、

前記第1過程で検出した出現頻度が複数種類のデータで共通に高頻度となる共通部分列群（S_{00}）と各データ毎に高頻度となる部分列群とを合わせた固有部分列群（S_{10}，S_{20}）を抽出し、各固有部分列群（S_{10}，S_{20}）毎に特定の群番号（j＝1，2）を付けて辞書領域（A_1，A_2）を確保し、該辞書領域（A_1，A_2）に前記共通部分列群（S_{00}）を合せた各固有部分列群（S_{10}，S_{20}）に属する各部分列を初期登録する第2過程と、

複数種類のデータが混在する入力文字列を符号化する際に、入力文字列に最長一致する部分列を前記辞書から検索し、検索した部分列が前記共通部分列群（S_{00}）に属するときは該共通部分列群（S_{00}）内の登録番号（i）を用いて符号化し、一方、固有部分列群（S_{10}，S_{20}）に属するときは、該固有部分列群の群番号（A_1，A_2）と該群内の登録番号（i_j）とを用いて符号化する第3過程と、を備えたことを特徴とする複数種類データのデータ圧縮方法。

2

【請求項3】請求項1，2記載の複数種類データのデータ圧縮方法に於いて、前記第3過程では、符号化すべき入力文字列に最長一致する部分列が、前回符号化で最長一致した部分列と同じ部分列群に属しているときは、該群内の登録番号のみを用いて符号化し、前回符号化で最長一致した部分列と異なる部分列群に属しているときは、該群番号と該群内の登録番号を用いて符号化することを特徴とする複数種類データのデータ圧縮方法。

【請求項4】請求項1，2記載の複数種類データのデータ圧縮方法に於いて、前記第1過程では、出現する全種類のデータのサンプルごとに符号化を行って相異なる部分列の出現頻度を計数することを特徴とする複数種類データのデータ圧縮方法。

【請求項5】請求項1，2記載の複数種類データのデータ圧縮方法に於いて、前記第3過程にあっては、入力文字列を辞書の最長一致する部分列の検索で符号化した際に、該符号化済み文字に次の入力一文字を加えた文字列を、符号化文字列が属する部分列群に新たな参照番号を付けて登録することを特徴とする複数種類データのデータ圧縮方法。

【請求項6】請求項1，2記載の複数種類データのデータ圧縮方法に於いて、前記第2過程にあっては、各部分列群ごとに最大登録個数を予め定めて該部分列群を登録するメモリ領域を割り当てておき、前各群に属する部分列の登録番号を各メモリ領域の先頭からの位置で表すことを特徴とする複数種類データのデータ圧縮方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、ＬＺＷ符号等の動的辞書型アルゴリズムを用いて入力データを圧縮符号化する複数種類データのデータ圧縮方法に関する。近年、文字コード，ベクトル情報，画像など様々な種類のデータがコンピュータで扱われるようになっており、扱われるデータ量も急速に増加してきている。

【0002】大量のデータを扱うときは、データの中の冗長な部分を省いてデータ量を圧縮することで、記憶容量を減らしたり、速く伝送したりできるようになる。様々なデータを１つの方式でデータ圧縮できる方法としてユニバーサル符号化が提案されている。ここで、本発明の分野は、文字コードの圧縮に限らず、様々なデータに適用できるが、以下では、情報理論で用いられている呼称を踏襲し、データの１ワード単位を文字と呼び、データが任意ワードつながったものを文字列と呼ぶことにする。

【0003】ユニバーサル符号の代表的な方法として、ジブーレンペル（Ziv-Lempel）符号がある（詳しくは、例えば、宗像『Ziv-Lempelのデータ圧縮法』、情報処

3

理、Vol.26,No.1,1985年を参照のこと）。
ジブーレンペル符号では
①ユニバーサル型（スライド辞書型）と、
②増分分解型（Incremental parsing ；動的辞書型）
の２つのアルゴリズムが提案されている。

【０００４】さらに、ユニバーサル型アルゴリズムの改良として、ＬＺＳＳ符号、（T.C. Bell, "Better OPM/L Text Compression ", IEEE Trans. on Commun., Vol.COM-34, No.12 Dec.1986 参照）。また、増分分解型アルゴリズムの改良としては、ＬＺＷ（Lempel-Ziv-Welch）符号がある（T.A.Welch, "A Technique for High-Performance Data Compression ", Computer, June 1984 参照）。

【０００５】これらの符号の内、高速処理ができることと、アルゴリズムの簡単さからＬＺＷ符号が記憶装置のファイル圧縮などで使われるようになっている。

【０００６】
【従来の技術】従来のＬＺＷ符号の符号化アルゴリズムを図１５のフローチャートに示す。ＬＺＷ符号化は、書き替え可能な辞書をもち、入力文字列を相異なる部分列に分け、この部分列を出現した順に参照番号を付けて辞書に登録するとともに、現在入力している文字列を辞書に登録してある最長一致する部分文字列の参照番号だけで表して、符号化するものである。

【０００７】尚、増分分解型符号およびＬＺＷ符号の技術は、特開昭５９－２３１６８３号、米国特許第４，５５８，３０２号，米国特許第４，８１４，７４６号で開示されている。図１５の符号化処理は次のようになる。
ステップＳ１；予め全文字につき一文字からなる文字列を初期値として登録してから符号化を始める。辞書の登録数ｎを文字種数Ａと置く。

【０００８】カーソルをデータの先頭の位置に置く。
ステップＳ２；カーソルの位置からの文字列に一致する辞書登録の最長文字列Ｓを見つける。
ステップＳ３；文字列Ｓの識別番号を［$\log_2 n$］ビットで表して出力する。但し、［ｘ］はｘ以上の最小の整数である。辞書登録数ｎを一つインクリメントする。

【０００９】ステップＳ４；文字列Ｓにカーソルの最初の文字Ｃを付加した文字列ＳＣを辞書に登録する。カーソルはＳの後の文字に移動させる。
ステップＳ５；入力データの終了をチェックし、終了するまでステップＳ２～Ｓ４の処理を繰り返す。
図１６は従来のＬＺＷ復号化のアルゴリズムを示したフローチャートであり、図１５の符号化アルゴリズムと逆の操作を行って入力符号から文字列を復元し、同時に辞書を作成する。

【００１０】このような従来のＬＺＷ符号では、複数の異なる性質をもつ複数種類のデータが混在するデータを符号化すると、複数種のデータに合わせた辞書が作成されて符号化が行なわれる。複数種が混在するデータとしては、例えば、文字コードと画像が混在するデータが挙げられる。辞書のサイズが十分大きいときは、出現した全てのデータ種を含む辞書が作られるため、個々のデータ種単独で圧縮した場合に比べて圧縮率が悪化するという問題がある。

4

【００１１】また、辞書のサイズが小さいために１種類のデータ分しか登録できないようなきは、各データの種類が著しく異なれば、辞書をクリアして再学習するため、個々のデータ種に合わせた辞書が作られ、圧縮率は低下しない。しかし、データ中に同じデータ種が交互に出現するときは、そのつど学習し直すため、圧縮率が高められないという問題点があった。

【００１２】この問題点を解決するため、本願発明者らは、データの種類ごとに辞書を分けて作成することで高い圧縮率を得るようにした方法を提案している。図１７にデータの種類毎に辞書を作成して符号化するＬＺＷ符号化アルゴリズムを示す。図１７のＬＺＷ符号化は次のようになる。

【００１３】ステップＳ１；データの種類ごとに高頻度で出現する文字列を求め、辞書の初期値とする。
ステップＳ２；データの種類ごとに初期値を分割辞書に設定する。カーソルを１とし、辞書アドレスn_1をデータｊの初期値の個数Ａｊとし、直前辞書番号をｐｐ＝０とする。

【００１４】ステップＳ３；カーソルをセットした位置からの人力文字列に一致する各辞書ｊ中の最長の文字列Ｓｊ＝Ｓ１，Ｓ２，・・・，Ｓｋを見つける。
ステップＳ４；ステップＳ３で見つけた文字列Ｓｊの中から最長の文字列Ｓｐを求める。

【００１５】ステップＳ５；現在辞書番号ｐと直前辞書番号ｐｐが一致するかどうか判定する。
ステップＳ６；辞書番号が不一致の場合は、［$\log_2 n_{pp}$］ビットを用いて辞書が変ったことを示す識別番号０を表わし、また［$\log_2 K$］ビットを用いて変った辞書番号ｐを表わして出力する。

【００１６】ステップＳ７；ステップＳ６の出力が済みまたはステップＳ５で辞書番号が一致した場合に、ステップＳ４で検索した文字部分列Ｓｐの番号を［$\log_2 n_p$］ビットを用いて表わし、出力する。辞書アドレスn_pを１つインクリメントする。
ステップＳ８；文字列Ｓの次の文字をＣにセットする。符号化済み文字列Ｓｐに文字Ｃを加えた文字列ＳｐＣを、辞書アドレスn_pで辞書に登録する。現在辞書番号ｐを直前辞書番号ｐｐに置き替える。

【００１７】カーソルを文字列Ｓの位置の文字に移動させる。
ステップＳ９；データ終了の有無を判別し、終了していなければステップＳ３に戻り、終了していれば一連の処理を終る。
図１８は図１７のＬＺＷ符号化アルゴリズムの変形を示

5

したもので、図16のステップS5，S6で行っている参照辞書が変化を示す情報の符号化出力を除いており、他の点は同じになる。

【0018】この図17，図18に示す複数種類データのデータ圧縮方法では、データの種類ごとに高頻度で出現する文字列を調べて、データの種類ごとの辞書Diに予め設定しておき、複数個の辞書Diから検索した最長一致文字列の中から最も一致長が長い文字列の辞書を選んで符号化するものである。このため高頻度の初期値を元にデータの種類が分類され、データ種に適する辞書Diが選ばれるため、高圧縮率を得ることができる。

【0019】

【発明が解決しようとする課題】しかしながら、複数種類のデータが混在するデータを図17，図18の方法で符号化する場合、高圧縮率は得られるものの、複数個の辞書について最長一致する文字列を検索しなければならず、辞書検索に時間がかかるという問題がある。この辞書検索の問題は、ハードウェアで並列処理を行うようにすれば単一辞書を用いた従来のLZW符号化と同等の処理速度が得られるが、ソフトウェアによるシーケンシャル処理では辞書の複数の個数分の検索時間がかかり、処理速度が低下する問題があった。

【0020】本発明は、このような問題点に鑑みてなされたもので、複数種類のデータを対象に調べた高頻度に出現する文字列を初期登録した場合の符号化における辞書探索を一本化し、処理時間を増加させることなく高い圧縮率が得られるようにした複数種類データのデータ圧縮方法を提供することを目的とする。

【0021】

【課題を解決するための手段】図1は本発明の原理説明図である。まず本発明は、複数種類のデータが混在する入力データを符号化して圧縮する複数種類データのデータ圧縮方法として、次のようにする。

［第1過程］複数種類のデータをLZW符号化し、この符号化における辞書に登録した文字列の出現頻度を検出する。

【0022】［第2過程］第1過程のLZW符号化で検出した出現頻度が、複数種類のデータで共通に高頻度となる部分列を共通部分列群S_{00}として抽出し、共通部分列群S_{00}に特定の群番号j＝0を付けて辞書領域A_0を確保し、この辞書領域A_0に共通部分列群S_{00}に属する各部分列を初期登録する。

【0023】また第1過程のLZW符号化で検出した出現頻度が、複数種類のデータ毎に高頻度となる部分列を固有部分列群S_{10}，S_{20}として抽出し、各固有部分列群S_{10}，S_{20}毎に特定の群番号j＝1，j＝2を付けて辞書領域A_1，A_2を確保し、辞書領域A_1，A_2に該当する固有部分列群S_{10}，S_{20}に属する各部分列を初期登録する。

【0024】［第3過程］複数種類のデータが混在する

6

入力文字列を符号化する際に、入力文字列に最長一致する登録済み部分列を辞書10から検索し、検索した部分列の属する部分列群の群番号jと該部分列群内での検索文字列の登録番号iとを用いて入力文字列を符号化する。

【0025】また本発明の他の複数種類データのデータ圧縮方法としては、辞書領域を各データに共通な領域を各データに固有な領域と一緒にしてもよい。この場合の処理は、次のようになる。

［第1過程］複数種類のデータをLZW符号化し、この符号化における辞書に登録した文字列の出現頻度を検出する。

【0026】［第2過程］第1過程のLZW符号化で検出した出現頻度が複数種類のデータで共通に高頻度となる共通部分列群S_{00}と各データ毎に高頻度となる部分列群S_{10}，S_{20}とを合わせた固有部分列群（$S_{00}+S_{10}$，$S_{00}+S_{20}$）を抽出し、各固有部分列群（$S_{00}+S_{10}$，$S_{00}+S_{20}$）毎に特定の群番号j＝1，2を付けて辞書領域A_1，A_2を確保し、該辞書領域に各固有部分列群に属する各部分列を初期登録する。

【0027】［第3過程］複数種類のデータが混在する入力文字列を符号化する際に、入力文字列に最長一致する部分列を辞書から検索し、検索した部分列が共通部分列群S_{00}に属するときは共通部分列群内の登録番号nを用いて符号化し、一方、固有部分列群S_{10}，S_{20}に属するときは、固有部分列群の群番号j＝1，2と該群内の登録番号n_iとを用いて符号化する。

【0028】

【作用】このような本発明の複数種類データのデータ圧縮方法にあっては、予め出現するデータ種が例えば2種類の場合を例にとると、辞書に初期登録する文字列を2種類のデータについて高頻度で共通に現れる文字列と、各種類のデータだけに高頻度で現れる文字列との3つの集合S_{00}，S_{10}，S_{20}（文字列群）に分けて辞書に格納しておき、符号化時に入力データと最長一致する辞書中の文字列を、対応する集合ごとの参照番号j＝0，1，2を付けて符号化する。

【0029】これにより複数種類のデータ毎に調べた高頻度の文字列を単一の辞書に初期登録して複数種類のデータが混在するデータの符号化を行うことができ、ソフトウェアによる符号化であっても辞書検索を通常のLZWと同等の処理速度で行って高い圧縮率を得ることができる。

【0030】

【実施例】図2は複数種類のデータが混在したデータを符号化する本発明の複数種類データのデータ圧縮方法を実現する装置構成の一実施例を示した実施例構成図である。図2において、16はCPUであり、CPU16に対してはプログラムメモリ18とデータメモリ30が接続される。プログラムメモリ18にはコントロールソフ

ト２０，符号化ソフト２２，初期値作成手段としての機能を備えた辞書作成ソフト１４，出現頻度カウントテーブル２６及び出現頻度格納テーブル２８が設けられる。

【００３１】符号化ソフト２２は入力文字列に最長一致する辞書中の文字列を検索して辞書の参照番号を符号データとして出力する、例えばＬＺＷ符号化アルゴリズムを基本的に備える。また、復号化ソフト２４は符号化ソフト２２により符号化された入力符号列で辞書中の参照番号を検索し、対応する文字列を複号する例えばＬＺＷ復号化アルゴリズムを行う。

【００３２】辞書作成ソフト１４は符号化あるいは復号化に先立って行う初期値作成処理と符号化及び復号化の処理中に新たな文字列を辞書に登録する処理の２つを行う。この辞書作成ソフト１４における初期値作成機能はデータメモリ３０に格納された複数種類のデータを代表するサンプルデータ、例えば２種類のサンプルデータ１，２を対象に符号化ソフト２２に従った符号化を行い、この符号化時に辞書から文字を検索して符号データとして出力する毎に、符号データとして検索された辞書中の文字列の参照番号の使用回数を出現頻度カウントテーブル２６を使用してカウントアップし、文字列の出現頻度を検出する。

【００３３】サンプルデータの符号化が終了したならば出現頻度カウントテーブル２６の中のサンプルデータ１，２毎の出現頻度を参照し、２種類のデータ１，２に共通する高頻度の文字列の集合を初期値S_{00}として登録し、またデータ１，２のそれぞれで独立に高頻度に生ずる文字列の集合を初期値S_{10}，S_{20}として登録する。一方、データメモリ３０には辞書１０とデータバッファ３２の各メモリ領域が確保される。

【００３４】初期値作成時にはデータバッファ３２には初期値作成の対象となる複数種類のサンプルデータ、例えばサンプルデータ１，２が格納され、また辞書１０には初期値作成のための符号化時に辞書作成ソフト１４で作成された文字列が参照番号と共に登録される。初期値作成が済むと、辞書１０には辞書作成ソフト１４で作成された複数種類、例えば２種類のデータ１，２の共通の初期値S_{00}、各データ１，２毎の初期値S_{10}，S_{20}がそれぞれに割り当てられた領域A_0，A_1，A_2毎に初期登録が行われる。そしてデータバッファ３２には新たに符号化しようとするデータ１，２が混在した文字列あるいは復号化しようとする符号列が格納され、符号化ソフト２２による文字列の復号化あるいは復号化ソフト２４による文字列の復元が行われる。

【００３５】図３は本発明において２種類のサンプルデータ１，２を対象に辞書に登録する初期値の作成処理を示したフローチャートである。図３の初期値作成処理にあっては、まずステップＳ１で２種類のデータ１，２を対象にＬＺＷ符号化を行い、データ１，２に共通な高頻度の初期値S_{00}を作成する。

【００３６】続いてステップＳ２で共通の初期値S_{00}を辞書のA_{00}領域に格納してデータ１のみをＬＺＷ符号化し、データ１に特有の高頻度の初期値S_{10}を作成する。続いてステップＳ３で共通の初期値S_{00}を辞書のA_{00}領域に格納してデータ２のみをＬＺＷ符号化し、データ２に特有の高頻度の初期値S_{20}を作成する。具体的には、図４に示すようにサンプルデータ１，２を対象にＬＺＷ符号化を行って辞書に符号化済み文字列の部分列を参照番号と共に登録し、サンプルデータ１の符号化における出現頻度f_1とサンプルデータ２の符号化における出現頻度f_2のそれぞれを計数する。

【００３７】図５はサンプルデータ１，２の符号化で得られた出現頻度を縦軸にとり辞書の要素番号（参照番号）を横軸にとって示した説明図である。図５において、サンプルデータ１，２中の要素（文字列）の出現頻度をそれぞれf_1，f_2とし、また共通初期値S_{00}の閾値をT_0、各サンプルデータ１，２特有の閾値をT_1，T_2とすると、初期値S_{00}，S_{10}，S_{20}の集合は次のようになる。

初期値S_{00}：$f_1 > T_0$　かつ　$f_2 > T_0$ の要素の集合

初期値S_{10}：$f_1 \geqq T_0$　かつ　$f_2 \leqq T_0$　かつ　$f_1 > T_1$ の要素の集合

初期値S_{20}：$f_1 \leqq T_0$　かつ　$f_2 \geqq T_0$　かつ　$f_2 > T_2$ の要素の集合

このようにデータ１，２が混在した場合の符号化で作成される辞書要素の全体をデータ１，２に共通の集合S_{00}とデータ１，２に固有の集合S_{10}，S_{20}に分類して辞書に初期登録しておけば、この初期登録した辞書を用いた符号化で符号化中のデータが最長一致する辞書の参照番号がどの集合に属するかを調べることでデータ１，２の変移区間S_{00}なのか特定データ１，２の区間S_{10}またはS_{20}にあるかを判別することができ、単一の辞書を用いてデータの種類に対応した効率の良い符号化を行うことができる。

【００３８】図６は図３のステップＳ１に示したデータ１，２に共通の初期値S_{00}を作成する初期値作成処理を詳細に示したフローチャートである。図６において、まずステップＳ１にデータ１，２のそれぞれにおける全ての単一文字を初期値として登録してから符号化を始める。また辞書の登録数ｎを文字種数Ａと置き、カーソルをデータの先頭位置にセットし、更に出現頻度を計数するカウンタf_1を変移要素Ｎ個分準備して０に初期化する。

【００３９】次にステップＳ２でサンプルデータ１の入力を開始し、ステップＳ３でデータ入力の終了をチェックした後、ステップＳ４に進んでカーソル位置からの文字列に一致する辞書中の最長の文字列Ｓを見付ける。続いてステップＳ５で見付けた最長一致の文字列に含まれる全てのセット文字列について出現頻度f_1を１つイン

クリメントする。

【0040】次にステップS6で辞書アドレスnを1つインクリメントし、符号化した最長一致文字列Sの次の文字をCとし、この次の1文字を文字列Sに加えた文字列SCを参照番号nを付けて辞書に登録する。そして、カーソルを文字列Sの次の文字に移動させ、ステップS2で次のサンプルデータ1を入力する。ステップS2～S6の処理の繰返しにより、ステップS3でサンプルデータ1の入力終了が判別されるとステップS7に進み、再びカーソルを1にセットし、サンプルデータにおける出現頻度計数のため、サンプルデータ2の全要素分N個のカウンタf_2を0にリセットし、ステップS8でサンプルデータ2の入力を開始する。

【0041】続いてステップS9を介してステップS10に進み、サンプルデータ2のカーソル位置からの文字列に一致する辞書中の最長一致する文字列Sを見付け、ステップS11で見付けた最長一致の文字列に含まれる全てのセット文字列について出現頻度f_2を1つインクリメントする。続いてステップS12で辞書番号nを1つインクリメントし、検索した最長一致文字列Sの次の1文字をCとし、最長一致文字列S_1に次の1文字Cを加えた文字列SCを参照番号nを付けて辞書に登録し、カーソルを文字列Sの次の文字に移動させ、再びステップS8に戻る。

【0042】ステップS9でサンプルデータ2の入力終了が判別されるとステップS13に進み、サンプルデータ1の出現頻度f_1及びサンプルデータ2の出現頻度f_2が共に閾値T_0となる辞書中の文字列を取り出して初期値S_{00}とする。図7は図3の初期値作成処理におけるステップS2及びS3の詳細を示したフローチャートである。

【0043】図7にあっては、まずステップS1で図6で作成したサンプルデータ1，2に共通な高頻度の初期値S_{00}を辞書Dに格納し、カーソルを1に合わせ、辞書アドレスをn_0及びn_1にセットし、出現頻度を計数するカウンタfを0にリセットする。続いてステップS2でまずサンプルデータ1を入力し、ステップS3を介してステップS4で辞書中の最長一致する文字列Sを見つけ、ステップS4で最長一致文字列に含まれる全てのセット文字列について出現頻度fを1つインクリメントする。

【0044】続いてステップS6で辞書アドレスnを1つインクリメントし、最長一致文字列Sの次の1文字をCとし、この1文字を最長一致文字列Sに加えた文字列SCに参照番号nを付けて辞書に登録する。続いてカーソルを文字列Sの後ろの1文字に移動させ、ステップS2に戻って次のサンプルデータ1を入力する。以上のステップS2～S6の処理の繰返しによりサンプルデータ1の符号化が済むとサンプルデータ2の符号化に切り替わり、同様な処理を繰返す。

【0045】ステップS3でデータ入力の終了が判別されるとステップS7に進み、サンプルデータ1，2毎に計数されている出現頻度f_1，f_2に付き、閾値T_1，T_2以上となる辞書中の文字列を取り出してサンプルデータ1，2に特有な初期値S_{10}，S_{20}とする。図8はサンプルデータ1，2から作成された初期値S_{00}，S_{10}，S_{20}を用いた本発明によるLZW符号化アルゴリズムを示したフローチャートである。

【0046】図8において、まずステップS1において予めサンプルデータ1，2から作成した初期値S_{00}，S_{10}，S_{20}をそれぞれ辞書の領域A_{00}，A_{10}，A_{20}に格納する。また、各領域における既存の辞書登録の要素数n_0，n_1，n_2を設定する。図9は図8のLZW符号化で使用される辞書構成を示した説明図である。図9において、各符号は次の内容を示す。

A_0，A_1，A_2：共通部分、データ種1，データ種2の格納領域

N_{0max}，N_{1max}，N_{2max}：格納領域A_0，A_1，A_2の各格納領域の最大要数

S_{00}，S_{10}，S_{20}：共通部分，データ種1，データ種2の初期値

A_{00}，A_{10}，A_{20}：共通部分，データ種1，データ種2の初期値の格納領域

n_{00}，n_{10}，n_{20}：共通部分，データ種1，データ種2の初期値の要素数

A_{01}，A_{11}，A_{21}：共通部分，データ種1，データ種2の既登録要素の格納領域

n_0，n_1，n_2：共通部分，データ種1，データ種2の既登録要素数

A_{02}，A_{12}，A_{22}：共通部分，データ種1，データ種2の空き領域

例えば、データ1，2に共通な高頻度をもつ初期値S_{00}を登録した辞書領域A_0についてみると、初期格納領域A_{00}に初期要素数n_{00}の初期値S_{00}を登録している。この領域A_{00}に続いて初期値S_{00}を用いた符号化で新たに登録された要素を含む既登録要素領域A_{01}が設けられ、ここまでの既存の登録要素数をn_0としている。また領域A_0，A_1，A_2については最大要素数をN_{0max}，N_{1max}及びN_{2max}と予め定めている。

【0047】再び図8を参照するに、辞書に対する初期登録が済むとステップS2で入力データと最長一致する辞書中の文字列（要素）を探索し、参照番号iを求める。続いて参照番号iが含まれる辞書領域A_jより辞書領域番号jを求める。この実施例ではデータは2種類ではあることから辞書領域jは図9に示すようにA_0，A_1，A_2の3つであり、辞書領域番号jはj＝0，1，2のいずれかとなる。

【0048】次にステップS4で前回の辞書領域番号と今回求めた辞書領域番号jとが等しいか否かチェックし、等しければステップS6に進み、参照番号iを辞書

領域A_jに対応する番号i_jに変換して符号化出力する。一方、前回の辞書領域番号が今回求めた辞書領域番号jに一致しなかった場合にはステップS5で新たな辞書領域番号jを符号化してからステップS6で参照番号iの符号化出力を行う。

【0049】続いてステップS7で辞書領域A_jに空きがあれば、その辞書領域の辞書アドレスn_jを1つインクリメントし、最長一致した文字列に次の1文字を付加した文字列を辞書領域Aに参照番号n_jを付けて追加登録する。ステップS8ではデータ終了の有無をチェックしており、データが終了しなければステップS2に戻って同様な処理を繰り返し、データが終了すれば一連の符号化処理を終わる。

【0050】図8のステップS6における参照番号iを辞書領域A_jに対応する参照番号i_jに変換する処理は次のモード1～3に従って行う。

［モード1］

$0 \leqq i < N_{0max}$；辞書領域番号＝0

辞書領域の対応番号$i_0 = i + Np$

［モード2］

$N_{0max} \leqq i < N_{1max}$；辞書領域番号＝1

辞書領域の対応番号$i_1 = i - N_{0max} + Np$

［モード3］

$N_{1max} \leqq i$；辞書領域番号＝2

辞書領域に対応番号$i_2 = i - (N_{0max} + N_{1max}) + Np$

ここで、Npは予約語の数であり、この実施例では例えばNp＝5の予約語を辞書領域の先頭に設けている。例えば、図10に示すように、辞書の先頭アドレス0～4を予約語領域とし、この辞書アドレス即ち参照番号0～4を各予約領域に示した意味をもつ情報として使用する。

【0051】即ち、参照番号0は辞書領域番号A_0を示し、参照番号1は辞書領域A_1を示し、また参照番号2は辞書領域番号A_2を示す。また、参照番号3は辞書の初期化を指令する制御コマンドとしての意味をもつ。更に参照番号4は符号化データの終了を示すEOF等に用いる。このため、実際の辞書領域は予約語領域に続くアドレス5、即ち参照番号5から開始され、モード1～3に示すように検索した参照番号iに予約語数Npを加えることで実アドレスが求まる。

【0052】またモード1～3における辞書領域に対応した参照番号i_0，i_1，i_2は図9に示した辞書領域A_0，A_1，A_2における領域内での相対位置を示している。このため、絶対位置を示す参照番号iに対し各領域A_0～A_2内での相対位置を示す参照番号i_0，i_1，i_2に変換することで、より少ない数値の参照番号とでき、符号化データのビット長を短縮して圧縮率を高めることができる。

【0053】また、前記モード1～3に示すようにして求めた各領域の対応番号i_j＝i_0，i_1，i_2は各領域の要素数n_j＝n_0，n_1，n_2を用いて表現し得る最小ビット数である

［$\log_2$（n_j＋Np）］ビット

で符号化する。但し、［X］はX以上の最小の整数を示している。

【0054】更に図8のステップS5における辞書領域番号jの符号化にあっては、辞書領域番号jを

［$\log_2$（n_k＋Np）］ビット

で符号化することになる。図11は図8のLZW符号化で得られた符号化データの説明図であり、図11にあっては符号化に使用する辞書領域が領域A_1，A_0，A_2と変移していったときの符号化データを示す。

【0055】即ち、最初は辞書領域j＝1にあることから辞書領域番号j＝1を符号化し、続いて領域A_1に属する文字列の符号化データを出力する。符号化データを3つ出力すると4番目の符号化データは領域A_0に属していることから、ここで領域j＝0を符号データとした後に文字列の符号データを出力する。更に、符号化データが領域A_2に属すると領域j＝2を符号化して出力した後に領域A_2に属する文字列の参照番号の符号データを出力する。

【0056】図12は図8のLZW符号化アルゴリズムで得られた符号データから元の文字列を復元するLZW復号化アルゴリズムを示したフローチャートである。図12において、まずステップS1で図8の符号化と同様、初期値S_{00}，S_{10}，S_{20}をそれぞれ辞書の対応領域A_{00}，A_{10}，A_{20}に格納する。続いてステップS2で符号を入力し、ステップS3で辞書領域番号の符号入力の有無をチェックし、辞書領域番号があればステップS5で現在の辞書領域番号を更新してステップS2で本来の符号を入力する。

【0057】続いてステップS4で現在の辞書領域に対応する番号i_jである符号を、前述したモード1～3の対応番号を求める関係式を使用して辞書の参照番号iに戻す。次にステップS5で辞書を参照し、参照番号iに対応する文字列を復元する。ステップS6で前回の辞書領域A_kに空き領域があれば辞書アドレスn_kを1つインクリメントし、前回の復元した文字列に今回復元した文字列の先頭文字を付加した文字列を辞書領域A_kに辞書アドレスn_kを付けて登録する。

【0058】以上の処理をステップS7で全ての符号データの入力が済むまで繰り返し、符号データの入力がなくなれば処理を終了する。図13は本発明の第2実施例で使用する辞書構成を示した説明図である。即ち、図9に示す辞書構成の実施例にあっては、辞書をデータ1，2に共通の領域A_0とデータ1，2に特有な領域A_1，A_2に分けていたが、図13の実施例にあっては、データ1，2に共通な領域の各々とデータ1，2に特有な領域を一緒にして1つの辞書領域としたことを特徴とす

る。

【0059】即ち、図13の辞書構成にあっては、データ1，2に共通な高頻度の初期値S_{00}を登録した領域については、データ1，2に固有な領域A_1またはA_2の一部に含ませており、この共通の初期値S_{00}にデータ1，2に特有な領域A_1，A_2のそれぞれを加えた領域が実際の符号化に使用するデータ1，2に固有な辞書領域となる。

【0060】図13のように共通領域を各データに固有の領域と一緒にした場合の辞書構成におけるLZW符号化アルゴリズムは図8と同じになるが、図8のステップS6における最長一致した辞書の参照番号iを辞書領域に対応する番号i_1に変換するモード1～3の処理が異なる。図13の第2実施例における参照番号iを各領域の対応番号i_1に直す処理は次のモード1～3のようになる。

［モード1］

$0 \leqq i < N_{0max}$；辞書領域番号＝1または2

辞書領域の対応番号$i_0 = i + Np$

［モード2］

$N_{0max} \leqq i < N_{1max}$；辞書番号＝1

辞書領域の対応番号$i_1 = n_0 + i - N_{0max} + Np$

［モード3］

$N_{1max} \leqq i$；辞書番号＝2

辞書領域に対応番号$i_2 = n_0 + i - (N_{0max} + N_{1max}) + Np$

この第2実施例におけるモード1～3における対応番号i_0～i_2への変換は、モード1の共通領域A_0については最初の実施例と同じであるが、モード2，3については共通領域S_{00}の既登録要素数n_0分だけ領域A_1，A_2を拡張するように対応番号を求める。

【0061】図14は辞書参照番号iが領域A_1に属した場合の対応番号i_1への変換を示したもので、想像線で示す実際の辞書番号iに対する対応番号i_1を求めると、領域A_1の一部である共通領域A_0の既登録要素数n_0分だけ領域A_1を拡張した参照番号に変換することを意味する。これによって、領域A_1は共通領域A_0を含む1つの領域として扱われることになる。

【0062】その結果、入力データと最長一致する文字列が領域A_1またはA_2の一部である共通領域A_0に属するときは辞書領域番号j＝1，2の指定は不要となる。この共通領域A_{00}を各データ特有の領域A_{10}，A_{20}と一緒にした辞書構成による符号化は、結局は2つの辞書を切り換えて使用していることと同じになる。また、上記の実施例におけるLZW符号化にあっては、入力する混在データの統計的性質の変動も考慮し、共通初期値S_{00}の登録領域A_{00}，A_{10}，A_{20}に続いて空き領域A_{02}，A_{12}，A_{22}を設け、実際のLZW符号化で得られた新たな文字列を登録する学習領域とし、学習によって混在データの統計的性質の変動を吸収している。

【0063】しかしながら、入力データの統計的性質がデータの種類ごとに予め分かっていれば辞書の初期値登録領域に続く空き領域A_{02}，A_{12}，A_{22}は設けず、初期値だけで符号化を行っても良い。このように初期値S_{00}，S_{10}，S_{20}のみでLZW符号化を行った場合には辞書への登録操作が省略できるため、更に処理速度を向上させることができる。

【0064】更に、上記の実施例にあっては2種類のデータの符号化に適用した場合を例にとるものであったが、本発明はこれに限定されず、2種類以上のデータについても全く同様に適用することができ、この場合にはデータの種類毎に高頻度の共通部分と各データ固有の高頻度の部分とに分けて集合を作り、各集合毎に参照番号を割り振って符号化すれば良い。

【0065】

【発明の効果】以上説明したように本発明によれば、複数種類のデータについて調べた高頻度の出現文字列の初期値を1つの辞書に登録して複数種類の混在データの符号化及び復号化を行うことができ、単一辞書であることからソフトウエアによるシーケンシャル処理であってもデータの種類毎に分割辞書を用いた方法に比べ、より高速の処理を行って高圧縮率を得ることができる。

【図面の簡単な説明】

【図1】本発明の原理説明図

【図2】本発明の複数種類データのデータ圧縮方法を実施する装置構成の実施例構成図

【図3】本発明の初期値作成処理の概略を示したフローチャート

【図4】本発明における初期値作成処理の内容を示した説明図

【図5】本発明の初期値作成における符号化で得られたサンプルデータ1，2の出現頻度を示した説明図

【図6】図3の共通部分S_{00}の初期値作成アルゴリズムを示したフローチャート

【図7】図3の固有部分S_{10}，S_{20}の初期値作成アルゴリズムを示したフローチャート

【図8】本発明の第1実施例におけるLZW符号化アルゴリズムを示したフローチャート

【図9】図9の符号化で使用される辞書構成の説明図

【図10】図8の領域内の番号に変換する際に使用する予約語数Npの辞書内容を示した説明図

【図11】図8の符号化で得られる符号データの説明図

【図12】本発明の第2実施例におけるLZW符号化アルゴリズムを示したフローチャート

【図13】図12のLZW符号化で使用される辞書構成の説明図

【図14】図13における領域内の対応番号が意味する辞書領域の説明図

【図15】従来のLZW符号化アルゴリズムを示したフローチャート

15

【図１６】従来のＬＺＷ復号化アルゴリズムを示したフローチャート

【図１７】データの種類毎に調べて高頻度の文字列を分割辞書に初期登録して行うＬＺＷ符号化アルゴリズムを示したフローチャート

【図１８】図１７の変形を示したフローチャート

【符号の説明】

１０：辞書

１６：ＣＰＵ

16

１８：プログラムメモリ

２０：コントロールソフト

２２：符号化ソフト

２４：復号化ソフト

２６：出現頻度カウントテーブル

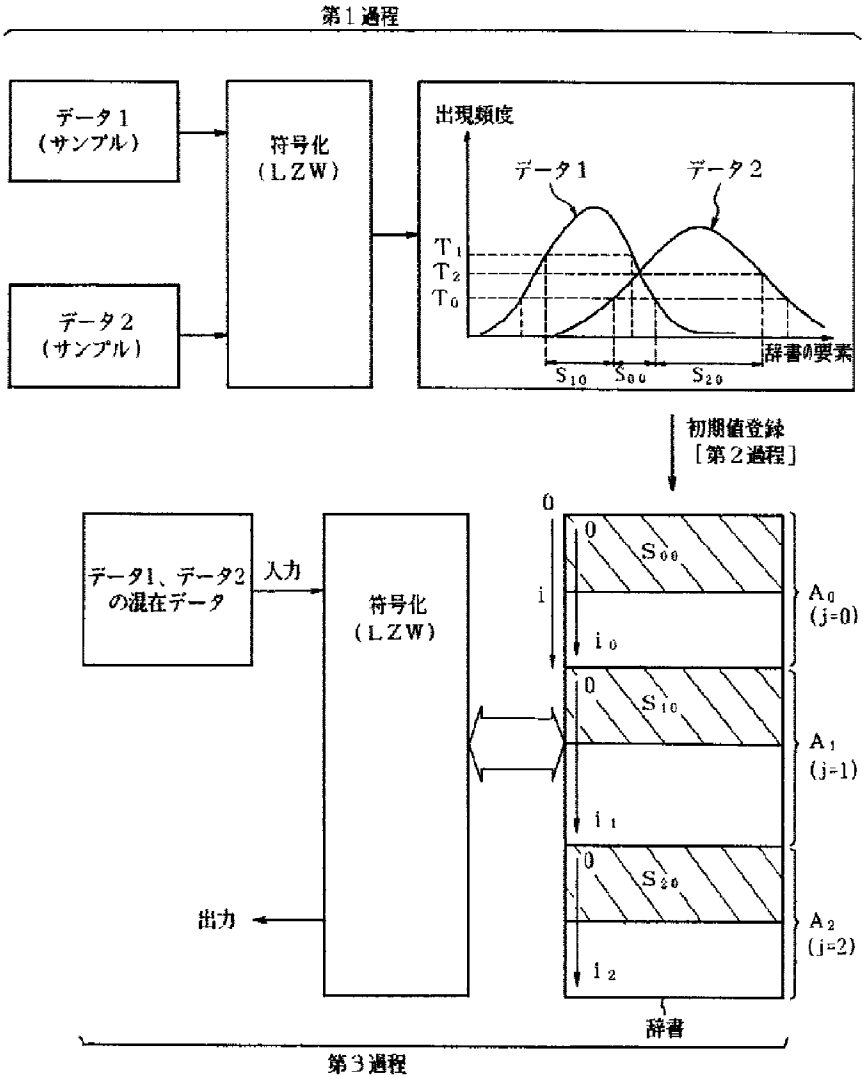
２８：頻度閾値格納テーブル

３０：データメモリ

３２：データバッファ

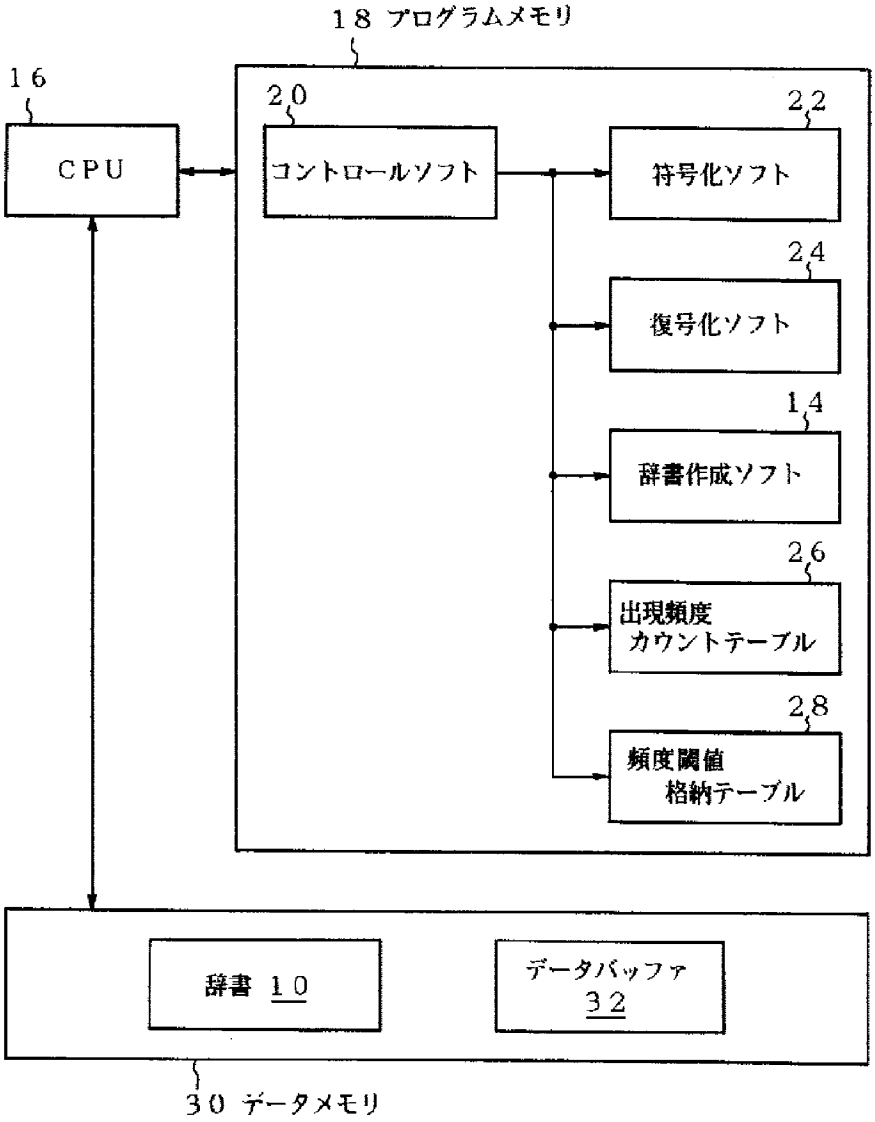
【図１】

本発明の原理説明図



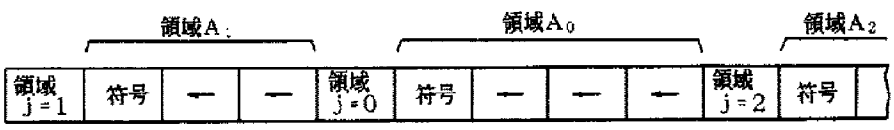
【図２】

本発明の複数種類データのデータ圧縮方法を実施する装置構成の実施例構成図



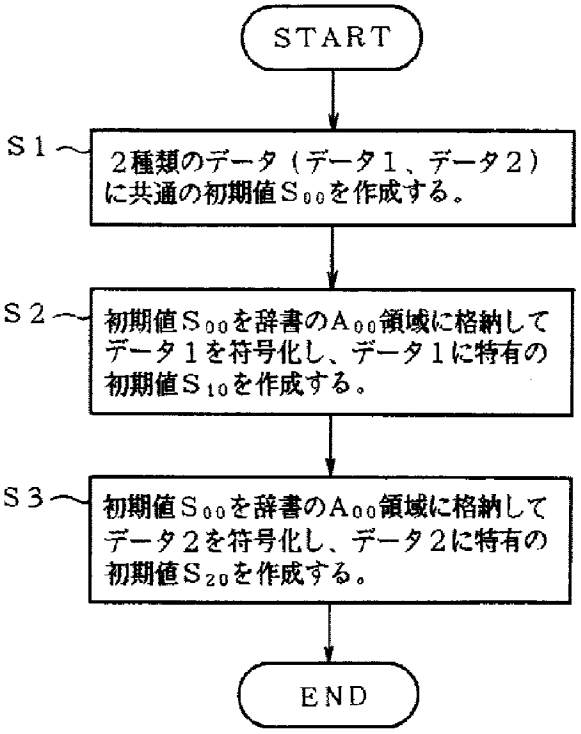
【図１１】

図８の符号化で得られる符号データの説明図



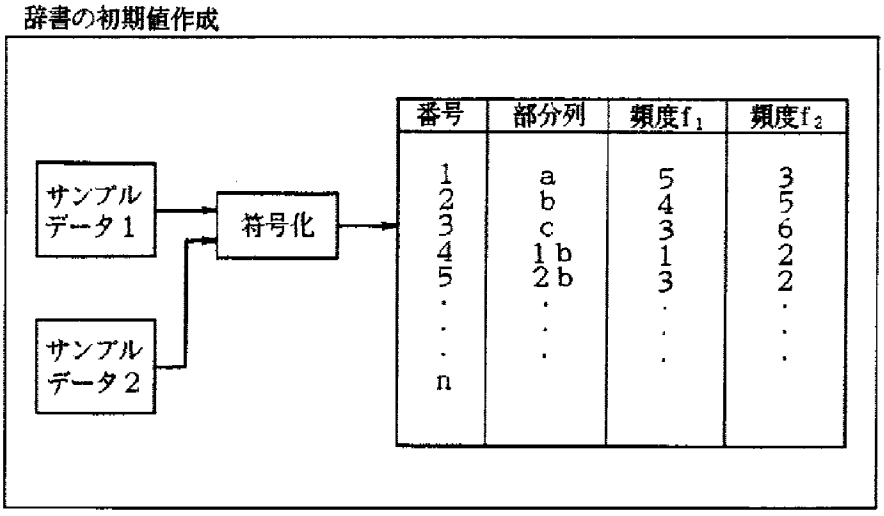
【図３】

本発明の初期値作成処理の概略を示したフローチャート



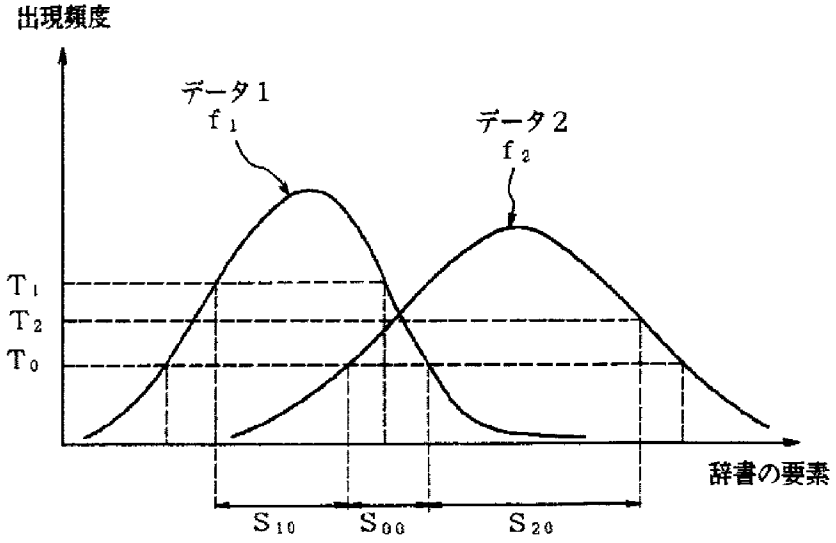
【図４】

本発明における初期値作成処理の内容を示した説明図



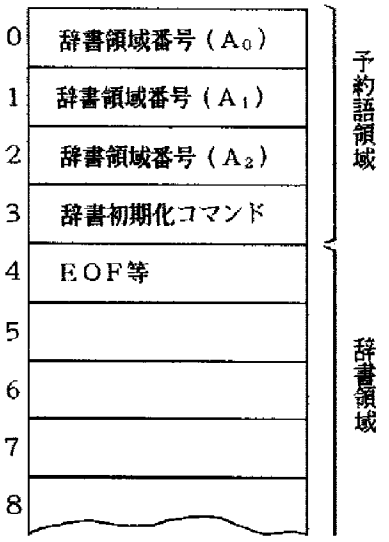
【図５】

本発明の初期値作成における符号化で得られたサンプルデータ１，２の出現頻度を示した説明図



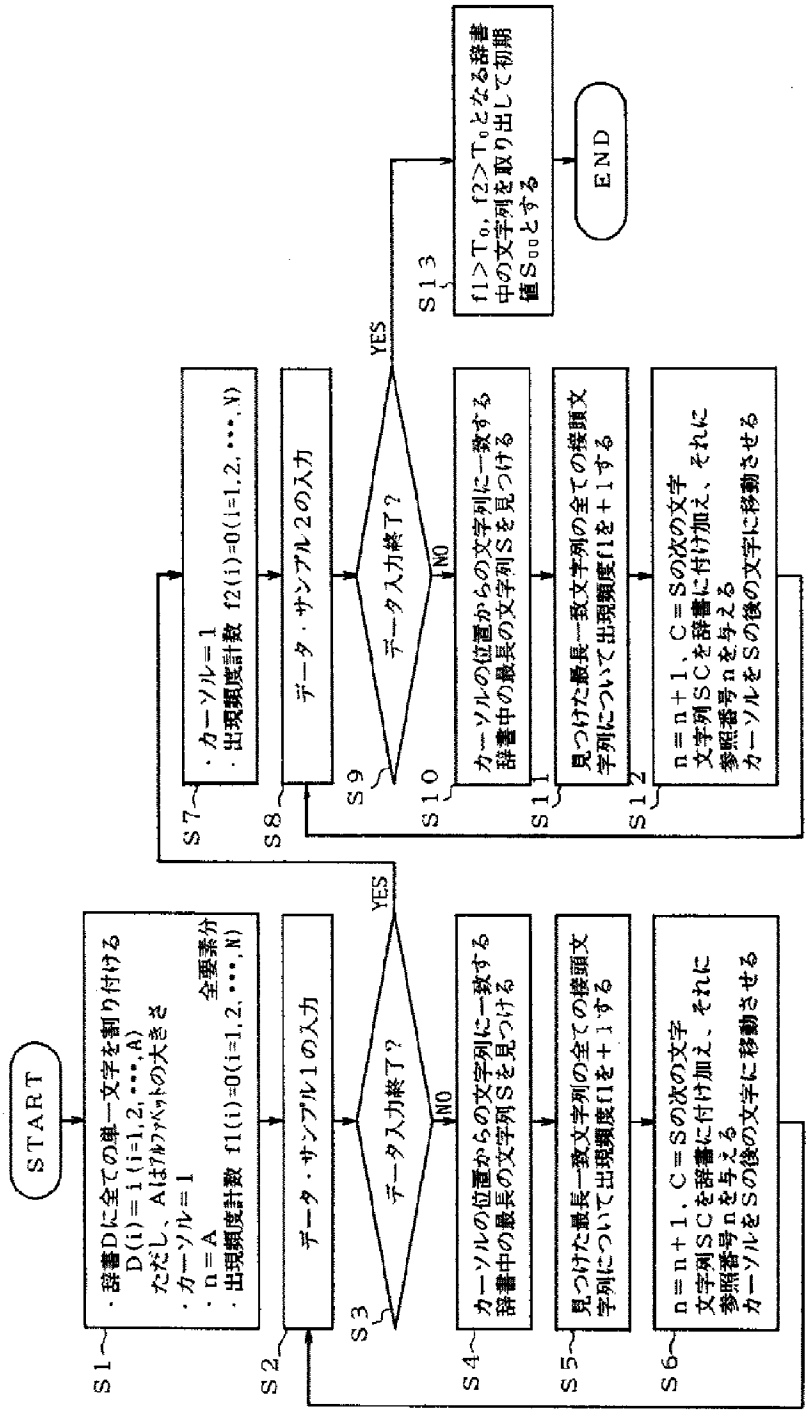
【図１０】

図８の領域内の番号に変換する際に使用する予約語数Ｎｐの辞書内容を示した説明図



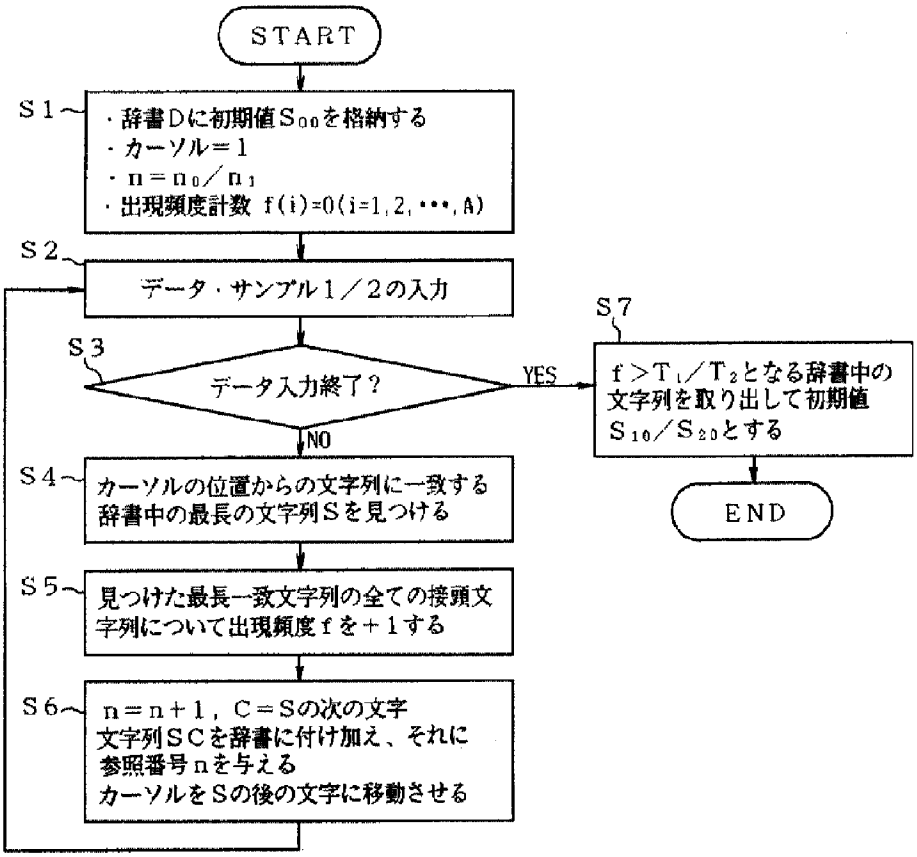
【図6】

図3の共通部分S_{oo}の初期値作成アルゴリズムを示したフローチャート



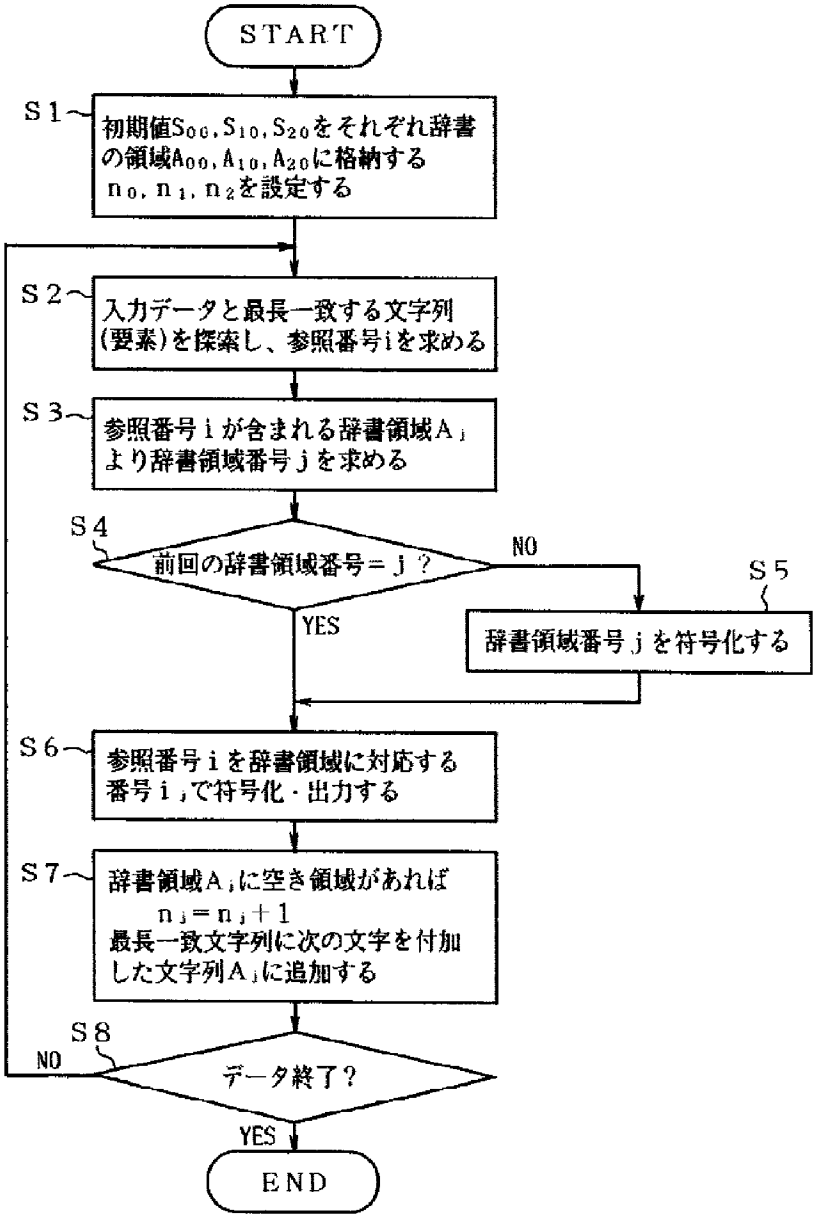
【図７】

図３の固有部分S_{10}，S_{20}の初期値作成アルゴリズムを示したフローチャート



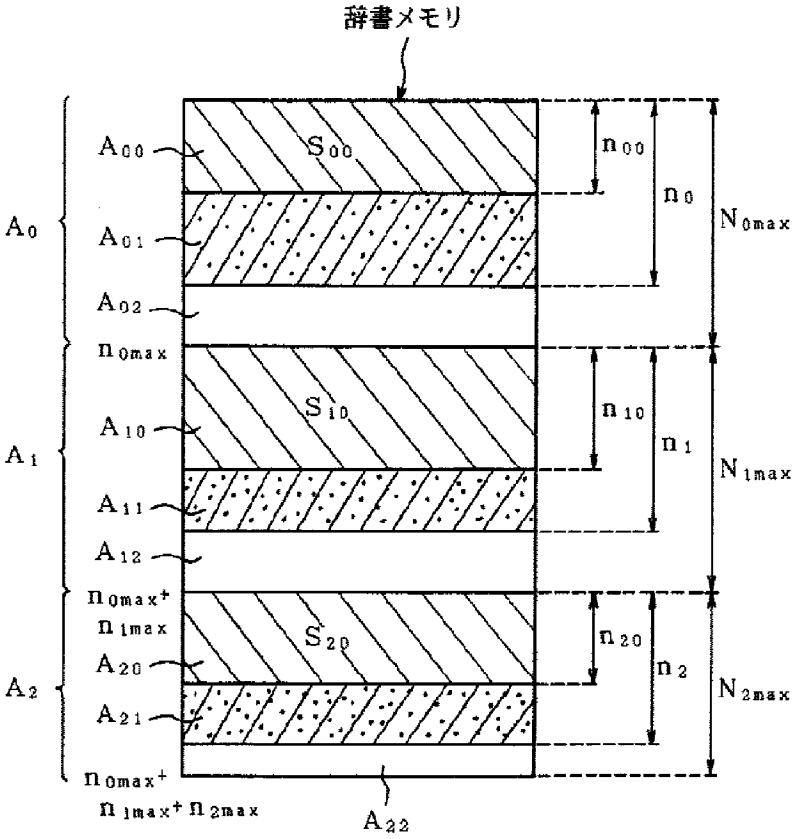
【図８】

本発明の第１実施例におけるＬＺＷ符号化アルゴリズムを示したフローチャート



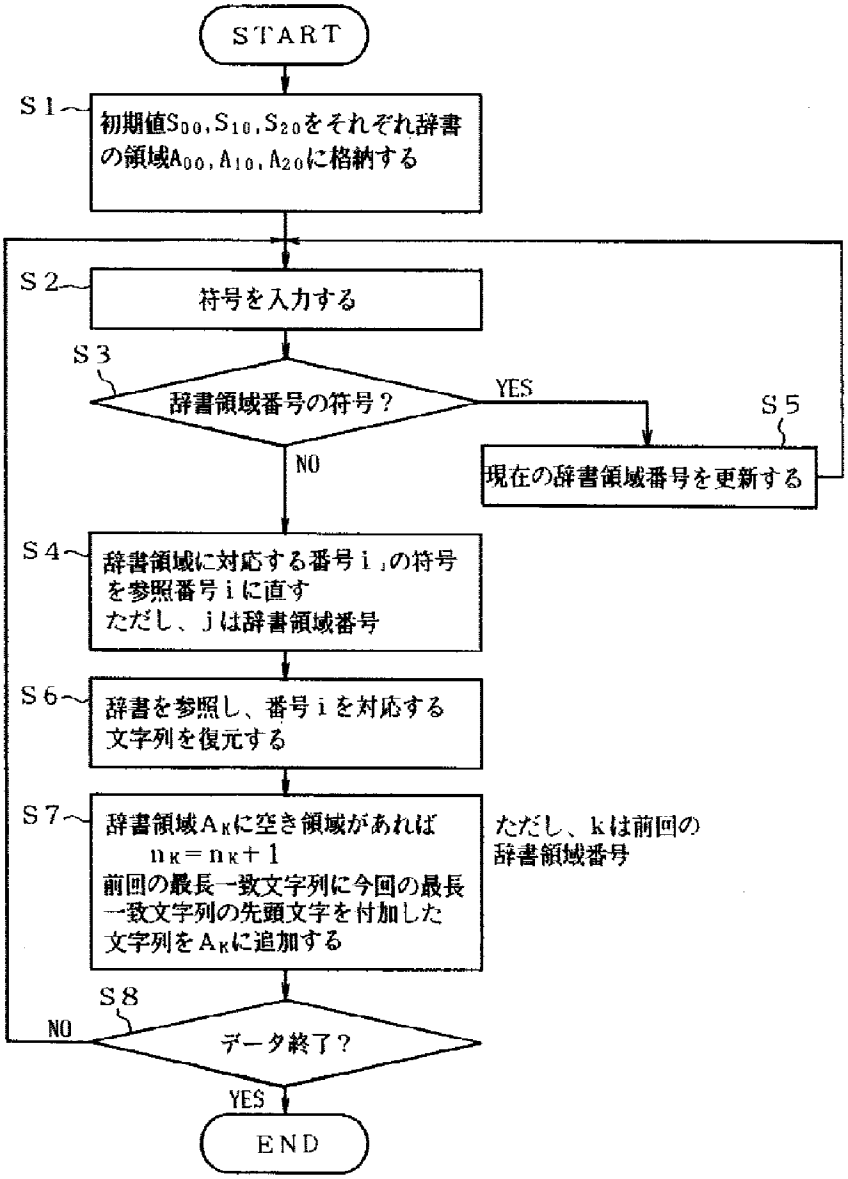
【図９】

図８の符号化で使用される辞書構成の説明図



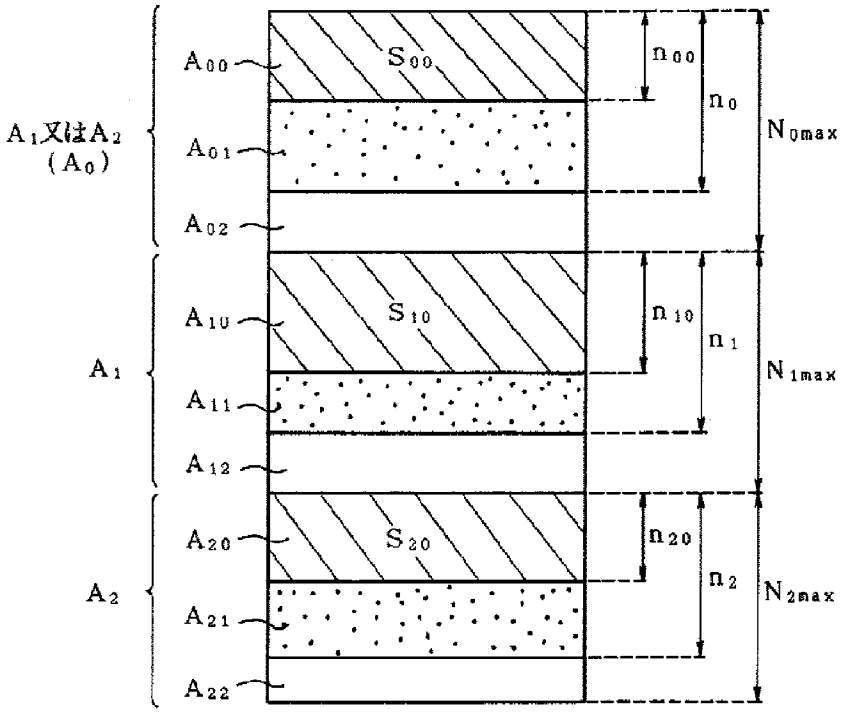
【図１２】

本発明の第２実施例におけるＬＺＷ符号化アルゴリズムを示したフローチャート



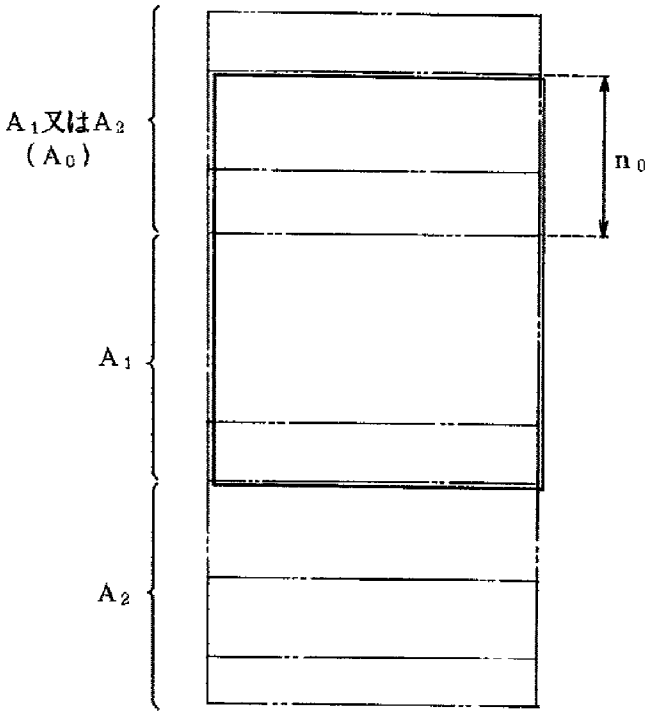
【図１３】

図１２のＬＺＷ符号化で使用される辞書構成の説明図



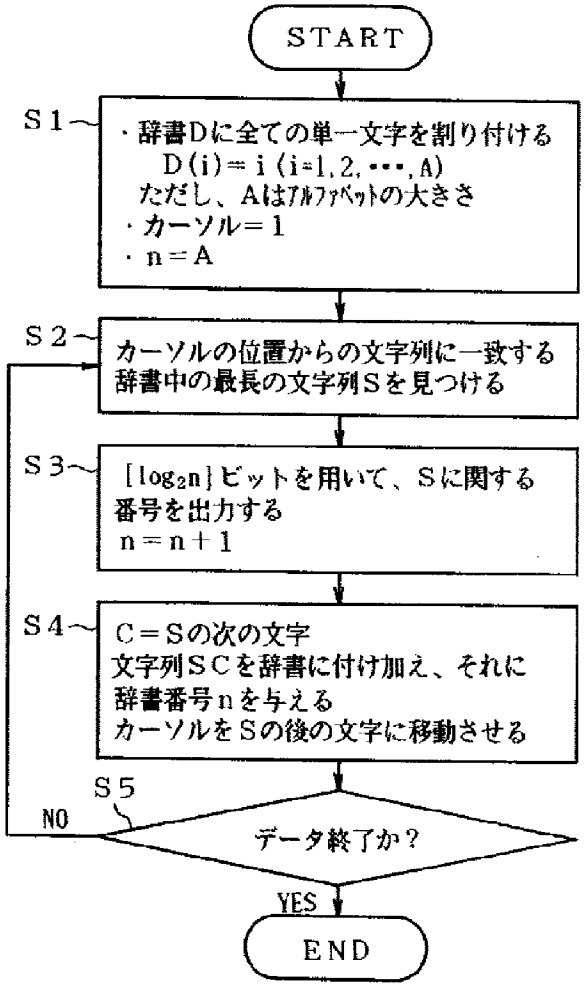
【図１４】

図１３における領域内の対応番号が意味する辞書領域の説明図



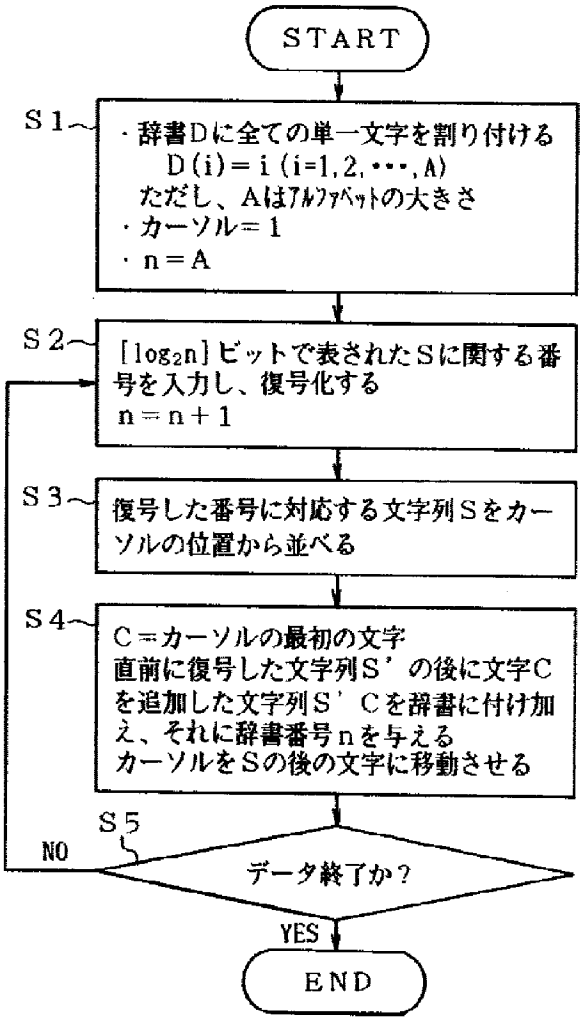
【図１５】

従来のＬＺＷ符号化アルゴリズムを示したフローチャート



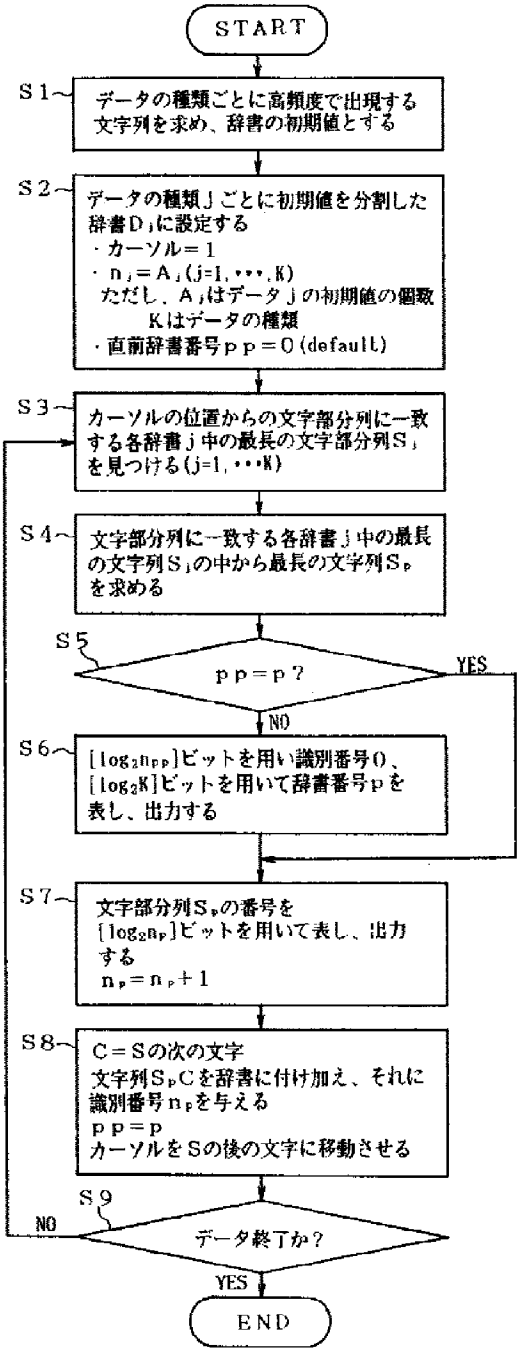
【図１６】

従来のＬＺＷ復号化アルゴリズムを示したフローチャート



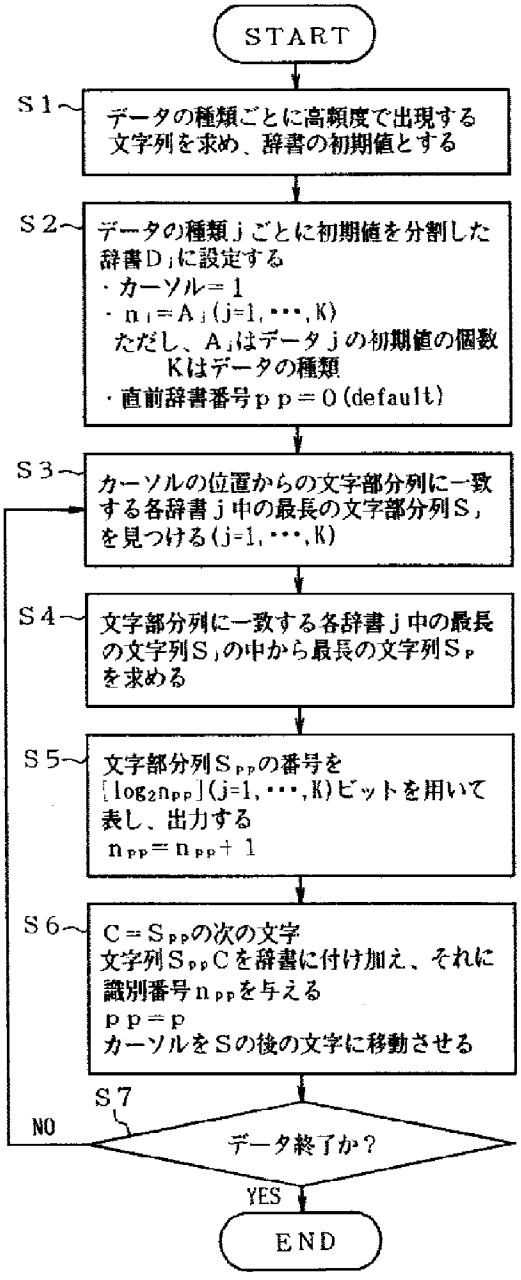
【図１７】

データの種類毎に調べて高頻度の文字列を分割辞書に初期登録して行うＬＺＷ符号化アルゴリズムを示したフローチャート



【図１８】

図１７の変形を示したフローチャート



フロントページの続き

(72)発明者　千葉　広隆
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内